(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Publication number: **0 378 422 A2**

(12) EUROPEAN PATENT APPLICATION

(21) Application number: 90300342.4

(51) Int. Cl.5: G06F 13/28

(22) Date of filing: 11.01.90

(30) Priority: 13.01.89 US 297715

(43) Date of publication of application: 18.07.90 Bulletin 90/29

(84) Designated Contracting States: DE FR GB

(71) Applicant: International Business Machines Corporation
Old Orchard Road
Armonk, N.Y. 10504(US)

(72) Inventor: Bailey, Roger Ned
6309 Pathfinder Drive
Austin, Texas, 78759(US)
Inventor: Mansfield, Robert Lockwood
12303 Meuse Cove
Austin, Texas, 78727(US)
Inventor: Spencer, Alexander Koos
12705 Cantle Trail
Austin, Texas, 78727(US)

(74) Representative: Bailey, Geoffrey Alan
IBM United Kingdom Limited Intellectual Property Department Hursley Park
Winchester Hampshire SO21 2JN(GB)

(54) Look ahead bus transfer request.

(57) A technique for use in an I/O channel to increase bus bandwidth during DMA data transfers between main system memory and a communication link is disclosed, including a pair of buffers, a plurality of counters adapted to selectively contain a count of data increments therein, and enhanced DMA control logic for monitoring buffer data content amount, and at a predetermined time during a given transfer initiating a bus arbitration so that it is completed simultaneously with the given transfer, thereby enabling the next data transfer to immediately commence.

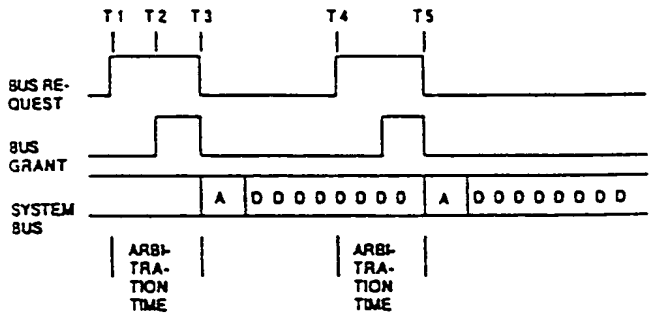

FIG. 3

EP 0 378 422 A2

buffers and several counters. A DMA controller monitors the state of DMA buffers and acts to fill or empty them when appropriate, based on the type of transaction in progress. A transmit circuit includes logic for removing data from the DMA buffers and placing it on the communications link. A receive circuit includes logic for taking data from the link and placing it into the DMA buffers.

A counter associable with each buffer during transmit operations is included in the DMA controller. Two counters are used to maintain the count of cache line units stored within the buffers during receive operations.

When receiving data from the link, the receive logic counts data as it is received and loads the cache line counter associated with each buffer with the number of cache lines received. The DMA controller transfers data over a shared I/O bus in cache line increments to main storage, decrementing the cache line counter on each line transfer until all current cache lines have been transferred.

While each transfer is in progress, the DMA controller determines if another cache line is to be transferred. If so, the controller then places a request on the bus before the transfer in progress is completed. Thus, the next transfer may begin as soon as the one in progress completes.

When transmitting data, the DMA controller reads from a tag word table which is resident in the channel. This table was loaded by the system processor with, inter alia, DMA addresses of data to be transmitted and the count of cache lines of data to be transmitted from those addresses.

The DMA controller then compares the count of cache lines in the tag table entry with the capacity of one of its pair of buffers and loads its cache line counter with the lesser of the total number of cache lines to be transmitted or the number of cache lines which may be accommodated in one buffer. The tag table cacheline count is decremented by the buffer capacity and if there is a positive remainder, that value replaces the original tag table cache line count. In this way the DMA controller will be able to know that more data remains to be transferred.

Data is retrieved from main system storage in cache line increments, and the DMA controller decrements its cache line counter until one of its pair of its buffers is filled.

If there is more data to be transmitted, the DMA controller fills the other of its buffers as above described while the first buffer is emptied onto the link by the transmit logic. The DMA controller iterates these steps until all data is transferred.

By using cache line counters to track when the associated buffers will be full, the DMA controller is able to prerequest the bus when necessary to enable back to back transfers. In this way the arbitration sequence occurs within transfer cycle time, increasing bus bandwidth.

Monitoring buffer status during transfer and communicating that status to independent data transfer circuits has several advantages. Multiple cache line transfers can be transferred between logic partitions without intervention between each. A bus request for a subsequent data transfer may be made before a current transfer is complete, pipelining arbitration into the transfer to better utilize available bus bandwidth to transfer data.

In order that the invention may be fully understood a preferred embodiment thereof will now be described, by way of example only, with reference to the accompanying drawings in which:-

Fig. 1 is a logical block diagram of an I/O channel in which the present invention is embodied, particularly illustrating elements of receive logic;

Fig. 2 is a logical block diagram similar to Fig. 1, illustrating elements of transmit logic;

Fig. 3 is a timing diagram of receive operation;

Fig. 4 is a timing diagram of transmit operation;

Fig. 5 is a flow chart of receive logic; and

Fig. 6 is a flow chart of transmit logic.

Refer now to Fig. 1 which is a schematic block diagram of an I/O channel. Serial data is received from a communications link over line 2. Receive control logic 4 counts received bytes in cache line increments and loads the result in either of cache line counters 8 and 10. Cache line counter 8 is associated with data buffer 14, and cache line counter 10 is associated with data buffer 18. In this exemplary, preferred embodiment a cache line increment is 64 bytes, and the capacity of each buffer 14 and 18 is 256 bytes. Therefore, cache line counters 8 and 10 contain values 0, 1, 2 or 3 to correspond to 64, 128, 192 or 256 bytes, respectively.

Counters 8 and 10 are accessible by DMA control logic 20. DMA control logic 20 reads a tag word from tag word table 22 which is provided by the system processor 24 with a plurality of entries, each including the main memory address ADDR which will receive the data from the link and a cache line count CLCNT, which for receive operations is set to some maximum value since the amount of data coming in from the link is unknown until an end of data indicator is received. This condition is made known to DMA control logic 20 over line 21.

When DMA control logic 20 determines from an interrogation of the active cache line counter 8 or 10 that either of buffers 14 and 18 is approaching the full state, or receives an end of data signal, it issues a request over bus request line 26. A bus

returns control to block 74.

When it is determined that CLCNT is 0, DMA logic queries the content status of buffer 18 at decision block 78 by examining counter 10. If buffer 18 is not full, the query sequence is repeated until the buffer is full. Then, at block 80 DMA control logic 20 again raises a bus request on line 26 and upon receipt of a grant over line 28 performs a cache line transfer. As before, the DMA CLCNT is decremented. The new value of DMA CLCNT is tested at block 82. While DMA CLCNT is greater than 0 as indicated at test 82, the data transfer counter decrement sequence is repeated. When DMA CLCNT goes to 0, the sequence beginning at query block 72 is repeated.

Fig. 6 is a flow chart of DMA control logic 20 during transmit operations and will be described having reference to Fig. 2 as well. Any given transmit operation begins as indicated at terminal 90. DMA control logic 20 at step 92 raises a bus request on line 26. Upon receipt of a grant on line 28, the address in main system memory is placed on the bus for accessing main memory. Data in a cache line increment is placed on the bus and loaded first into buffer 14. As before, the DMA CLCNT is decremented. At step 94 DMA control logic 20 determines if that buffer is full and continues the sequence of steps 92 and 94 until a full condition is indicated. When that condition occurs, control is passed at step 96 to transmit logic 50 for transmitting the data from buffer 14 to the communication link over serial line 54.

While transmit logic 50 empties buffer 14, DMA control logic begins loading buffer 18. This operation entails a bus request and accessing of main memory for placing the next cache line increment of data on the bus.

DMA control logic 20 then at step 98 determines whether buffer 18 is full. The bus request main memory access sequence just noted is repeated until buffer 18 is full. DMA control logic 20 then loops through step 100 until transmit logic 50 has completed the transmission of data from buffer 14. Then at step 102 DMA control logic 20 instructs transmit logic 50 to empty buffer 18 data onto line 52.

The entire sequence from step 92 through step 102 is repeated until the entire amount of data to be transmitted has been sent to the communications link.

Claims

1. A system for performing DMA block data transfers across a non-dedicated bus between a main memory and an external device comprising:
means for determining a total count of incremental data units comprising a block to be transferred;
means for transferring incremental data units;
means, associated with said means for transferring, for updating said total count of incremental data units to be transferred at each transfer; and
means for issuing a bus request for transfer of a subsequent incremental data unit before completion of transfer of a current incremental data unit.

2. A system for performing a DMA transfer of a block of data across a non-dedicated bus between a main memory and an external device, said block of data comprising a plurality of incremental data units comprising:
means for determining a total count of incremental data units comprising the block to be transferred;
means for issuing a bus request, to request access to the bus for the transfer of one or more incremental data units;
means for detecting a bus grant,
means, responsive to said detecting means, for transferring one or more incremental data units on said bus when said bus grant is detected;
means associated with said means for transferring, for updating at each transfer said total count of incremental data units still to be transferred; and
means for issuing a bus request, to request access to the bus for the transfer of one or more subsequent incremental data units, before completion of the current incremental data unit transfer.

3. A system as claimed in Claim 1 or Claim 2 additionally including:
buffer storage for a plurality of incremental data units.

4. A system as claimed in Claim 3 wherein, during data transfer from said external device to said main memory, said means for determining comprises:
in-counter means associated with said buffer storage; and
receive logic for counting incremental data units as they are received, loading said incremental data units in said buffer storage, and placing a total count of incremental data units received in said in-counter means.

5. A system as claimed in claim 4, wherein:
said means for transferring includes means for placing buffer storage contents on said non-dedicated bus; and
said means for updating comprises means for decrementing said in-counter means.

6. A system as claimed in any of Claims 3 to 5 wherein, during data transfer from said main memory to said external device, said means for determining comprises:
out-counter means adapted to accept a count of data increments from a processor controlling said main memory.

7. A system as claimed in Claim 6, wherein

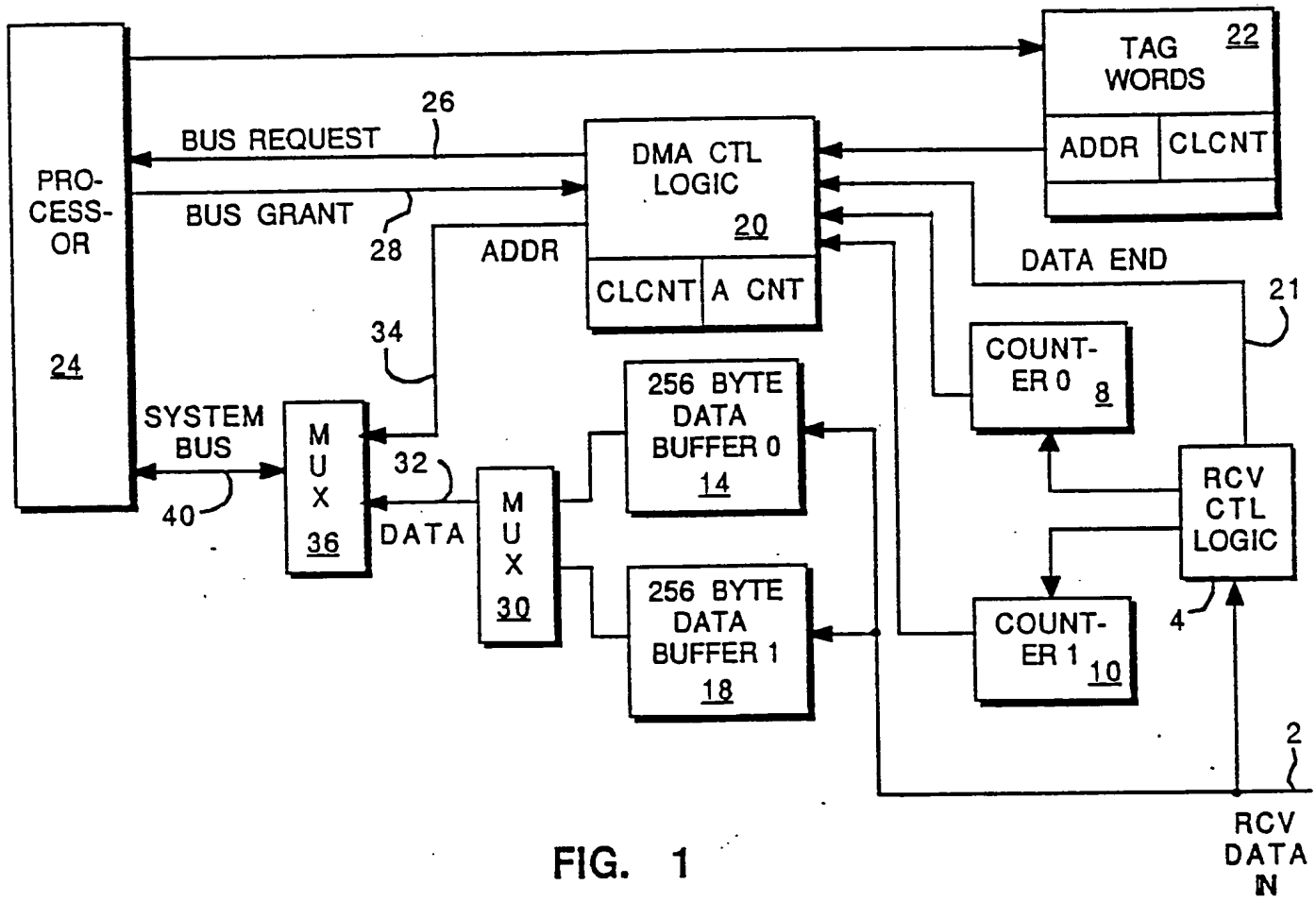
PRO-
CESS-
OR
24
BUS REQUEST
26
BUS GRANT
28
ADDR
34
DMA CTL
LOGIC
20
CLCNT
A CNT
TAG
WORDS
22
ADDR
CLCNT
DATA END
21
SYSTEM
BUS
40
M
U
X
36
32
DATA
M
U
X
30
256 BYTE
DATA
BUFFER 0
14
256 BYTE
DATA
BUFFER 1
18
COUNT-
ER 0
8
COUNT-
ER 1
10
RCV
CTL
LOGIC
4
2
RCV
DATA
IN

FIG. 1

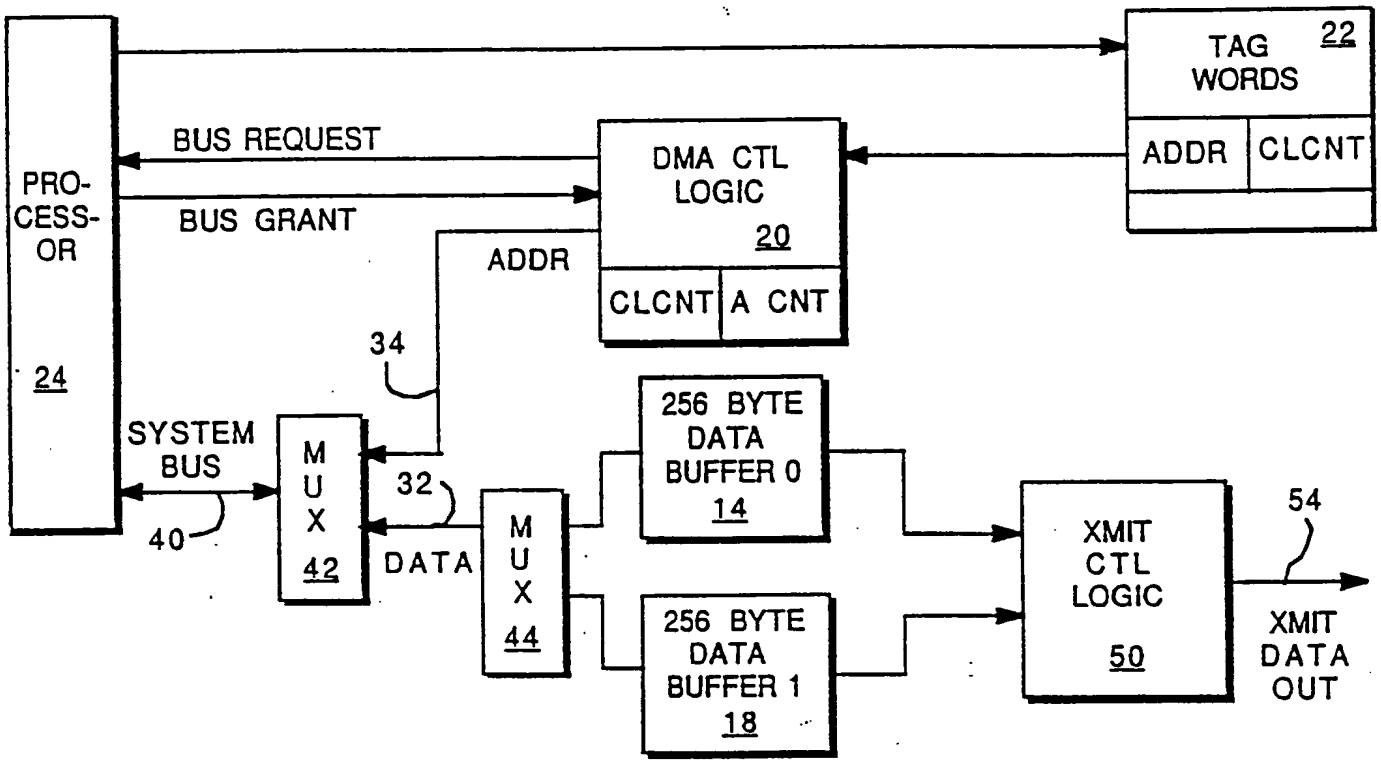
PRO-
CESS-
OR
24
BUS REQUEST
BUS GRANT
ADDR
34
DMA CTL
LOGIC
20
CLCNT
A CNT
TAG
WORDS
22
ADDR
CLCNT
SYSTEM
BUS
40
M
U
X
42
32
DATA
M
U
X
44
256 BYTE
DATA
BUFFER 0
14
256 BYTE
DATA
BUFFER 1
18
XMIT
CTL
LOGIC
50
54
XMIT
DATA
OUT

FIG. 2

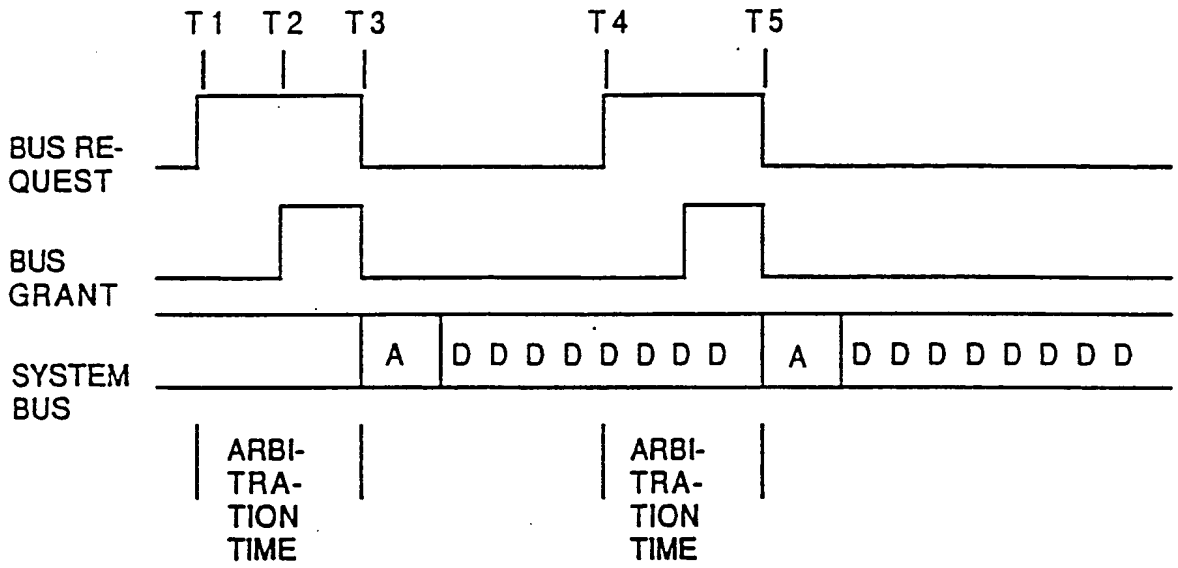
T1
T2
T3
T4
T5
BUS RE-QUEST
BUS GRANT
SYSTEM BUS
A
D D D D D D D D
A
D D D D D D D D
ARBI-TRA-TION TIME
ARBI-TRA-TION TIME

FIG. 3

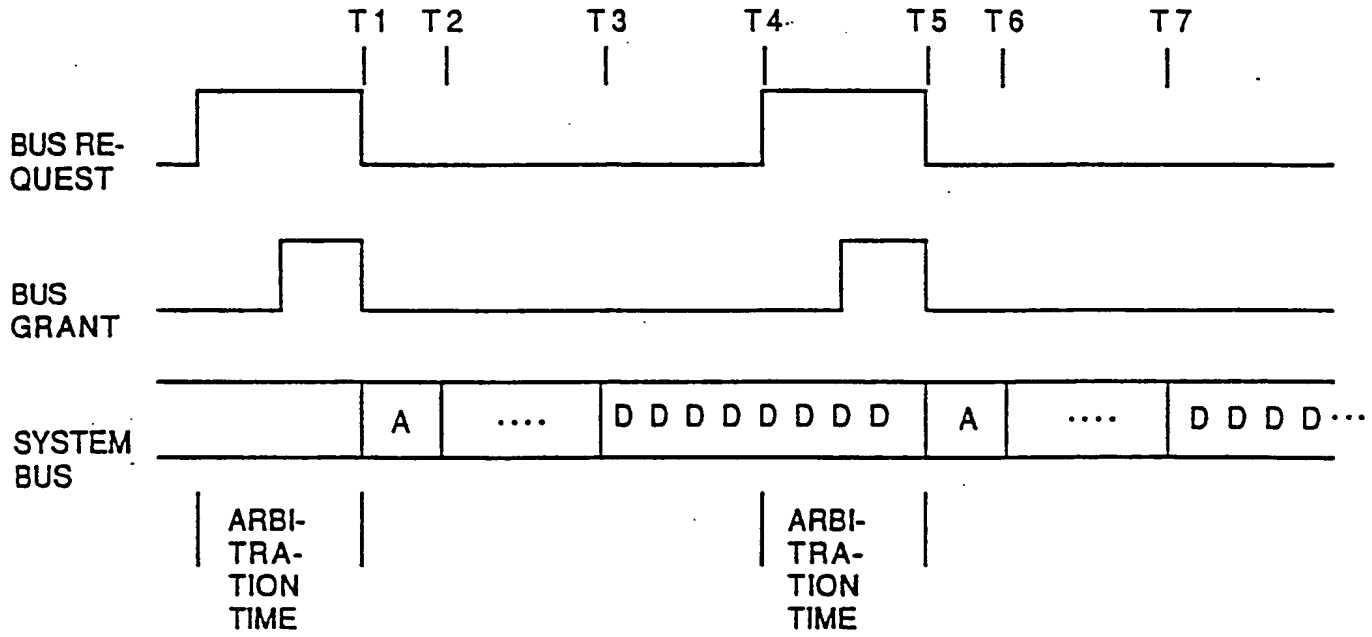
T1
T2
T3
T4
T5
T6
T7
BUS RE-QUEST
BUS GRANT
SYSTEM BUS
A
....
D D D D D D D D
A
....
D D D D ...
ARBI-TRA-TION TIME
ARBI-TRA-TION TIME

FIG. 4

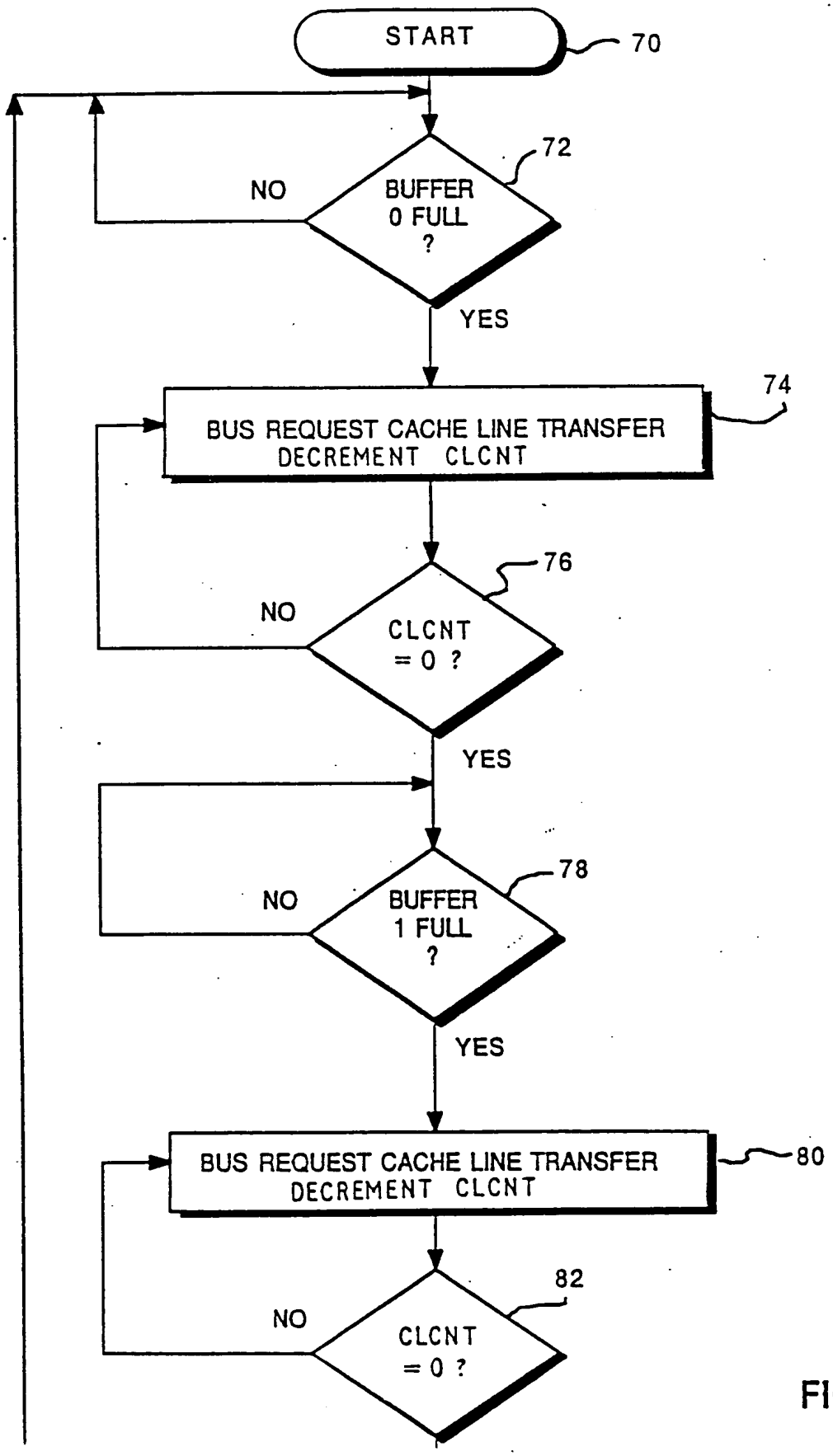
START
70
72
NO
BUFFER
0 FULL
?
YES
74
BUS REQUEST CACHE LINE TRANSFER
DECREMENT CLCNT
76
NO
CLCNT
= 0 ?
YES
78
NO
BUFFER
1 FULL
?
YES
BUS REQUEST CACHE LINE TRANSFER
DECREMENT CLCNT
80
82
NO
CLCNT
= 0 ?

FIG. 5

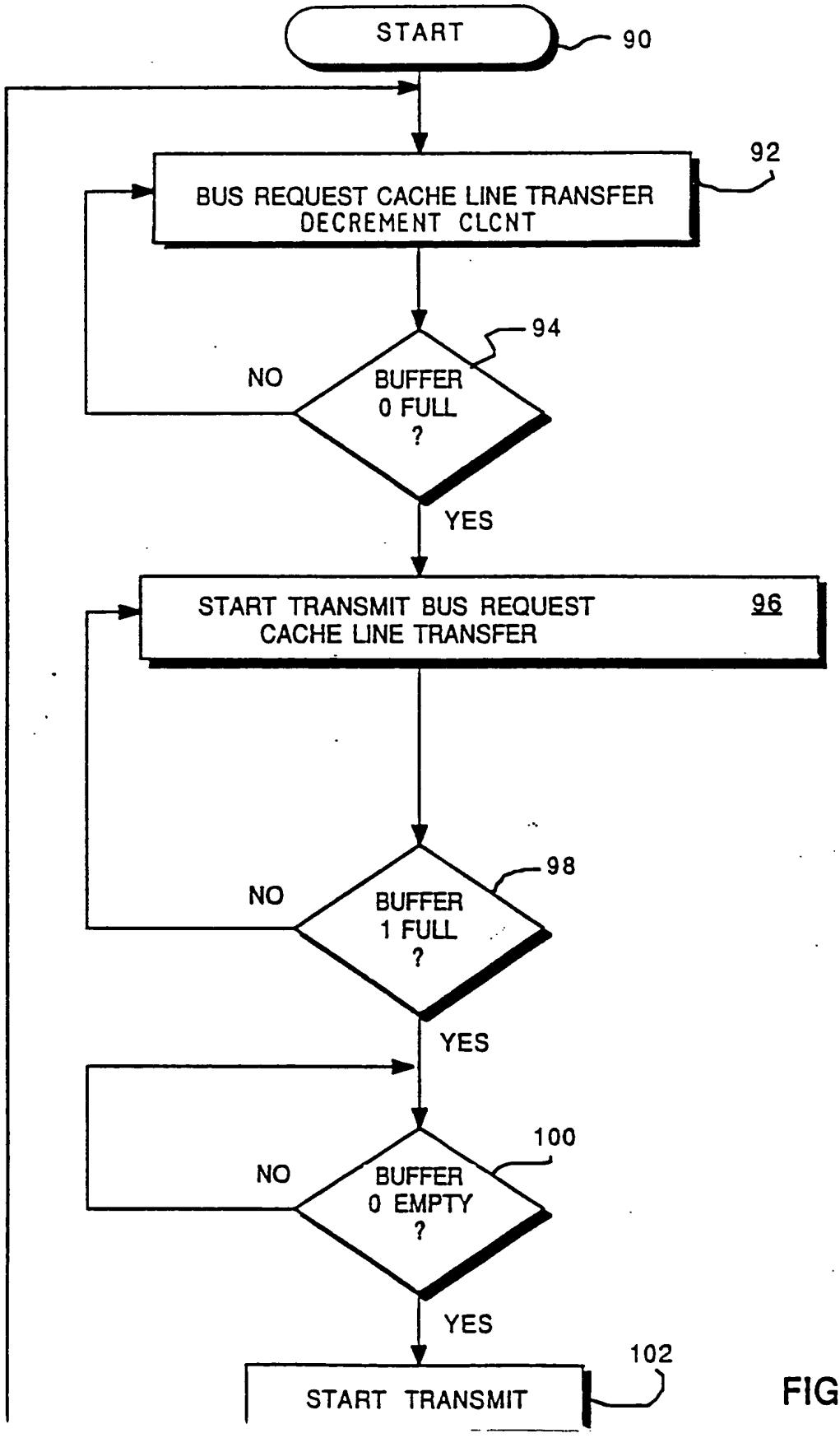
START
90
BUS REQUEST CACHE LINE TRANSFER
DECREMENT CLCNT
92
94
NO
BUFFER
0 FULL
?
YES
START TRANSMIT BUS REQUEST
CACHE LINE TRANSFER
96
98
NO
BUFFER
1 FULL
?
YES
100
NO
BUFFER
0 EMPTY
?
YES
102
START TRANSMIT

FIG. 6

(51) G06F13/28

(19) Office européen des brevets

(11) Publication number: **0 378 422 A3**

(12) EUROPEAN PATENT APPLICATION

(21) Application number: 90300342.4

(51) Int. Cl.5: G06F 13/28

(22) Date of filing: 11.01.90

(30) Priority: 13.01.89 US 297715

(43) Date of publication of application:
18.07.90 Bulletin 90/29

(84) Designated Contracting States:
DE FR GB

(88) Date of deferred publication of the search report:
29.05.91 Bulletin 91/22

(71) Applicant: International Business Machines Corporation
Old Orchard Road
Armonk, N.Y. 10504(US)

(72) Inventor: Bailey, Roger Ned
6309 Pathfinder Drive
Austin, Texas, 78759(US)
Inventor: Mansfield, Robert Lockwood
12303 Meuse Cove
Austin, Texas, 78727(US)
Inventor: Spencer, Alexander Koos
12705 Cantle Trail
Austin, Texas, 78727(US)

(74) Representative: Bailey, Geoffrey Alan
IBM United Kingdom Limited Intellectual Property Department Hursley Park
Winchester Hampshire SO21 2JN(GB)

(54) Look ahead bus transfer request.

(57) A technique for use in an I/O channel to increase bus bandwidth during DMA data transfers between main system memory and a communication link is disclosed, including a pair of buffers, a plurality of counters adapted to selectively contain a count of data increments therein, and enhanced DMA control logic for monitoring buffer data content amount, and at a predetermined time during a given transfer initiating a bus arbitration so that it is completed simultaneously with the given transfer, thereby enaoling the next data transfer to immediately commence.

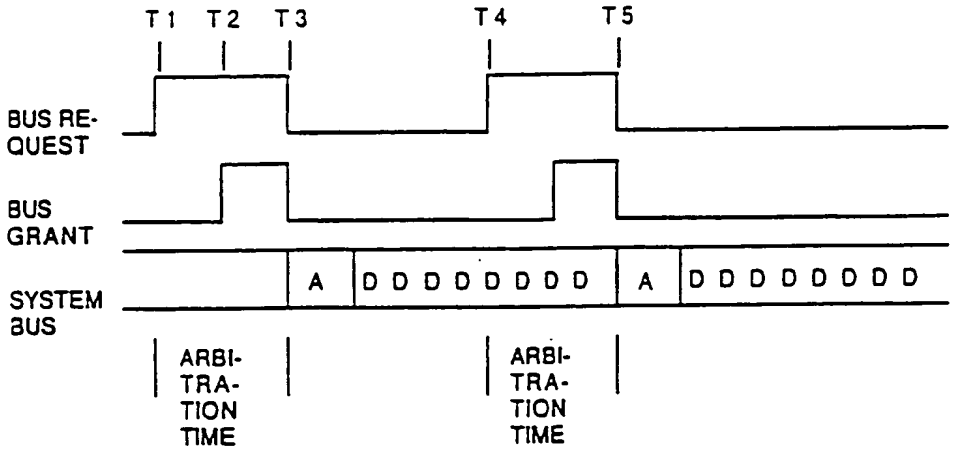

FIG. 3

EP 0 378 422 A3

European Patent Office

EUROPEAN SEARCH REPORT

Application Number

EP 90 30 0342

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int. Cl.5)
P,Y	EP-A-0 333 594 (FUJITSU) * Abstract; column 6, line 22 - column 7, line 38; figure 4 *	1-3,8-13	G 06 F 13/28
Y	US-A-4 797 809 (SATO et al.) * Abstract; column 1, line 35 - column 4, line 46; figure 1 *	1-3,8-13	
A	EP-A-0 109 308 (BURROUGHS CORP.) * Page 3, line 30 - page 5, line 10 *	1-13	
A	US-A-4 675 807 (GOURNEAU et al.) * Abstract; column 2, lines 14-61; figure 1 *	1-3,8	

TECHNICAL FIELDS SEARCHED (Int. Cl.5)

G 06 F 13

The present search report has been drawn up for all claims

Place of search	Date of completion of search	Examiner
The Hague	26 February 91	MCDONAGH F.M.

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same catagory
A : technological background
O : non-written disclosure
P : intermediate document
T : theory or principle underlying the invention

E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document